



①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ Übersetzung der  
europäischen Patentschrift

⑧7 EP 0 423 906 B1

⑩ DE 36 50 413 T 2

⑥1 Int. Cl.®:  
G 06 F 9/38

(11)

DE 36 50 413 T 2

- |    |   |              |
|----|---|--------------|
| ②1 | Deutsches Aktenzeichen:                               | 36 50 413.0  |
| ③6 | Europäisches Aktenzeichen:                            | 90 203 019.6 |
| ③6 | Europäischer Anmeldetag:                              | 12. 6. 86    |
| ③7 | Erstveröffentlichung durch das EPA:                   | 24. 4. 91    |
| ③7 | Veröffentlichungstag<br>der Patenterteilung beim EPA: | 4. 10. 95    |
| ④7 | Veröffentlichungstag im Patentblatt:                  | 7. 3. 96     |

③0 Unionspriorität: ③2 ③3 ③1  
28.06.85 US 750625

⑦3 Patentinhaber:  
Hewlett-Packard Co., Palo Alto, Calif., US

⑦4 Vertreter:  
Schoppe, F., Dipl.-Ing.Univ., Pat.-Anw., 82049 Pullach

⑧4 Benannte Vertragsstaaten:  
CH, DE, FR, GB, IT, LI, NL, SE

⑦2 Erfinder:  
Lee, Ruby Bei-Loh, Cupertino, California 95014, US;  
Baum, Allen J., Palo Alto, California 94306, US

⑤4 Verfahren und Vorrichtung zur Annullierung eines Befehls.

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 36 50 413 T 2

### Hintergrund

Die Stammanmeldung EP-A-0207665 bezieht sich auf ein Verfahren zur verzögerten bedingten Verzweigung und auf eine Vorrichtung zum Ausführen dieser verzögerten Verzweigung.

Die Fähigkeit, Entscheidungen durch eine bedingte Verzweigung zu treffen, ist eine essentielle Anforderung für jedes Computersystem, das eine sinnvolle Arbeit durchführt. Die Entscheidung zu verzweigen oder nicht zu verzweigen kann auf einem oder mehreren Ereignissen basieren. Diese Ereignisse, die oft als Bedingungen bezeichnet werden, schließen ein: positive, negative oder Null-Nummern, einen Überfluß, einen Unterfluß oder einen Übertrag von der letzten arithmetischen Operation, eine geradzahlige oder ungeradzahlige Parität, und viele andere. Bedingte Verzweigungen werden in digitalen Computern durch bedingte Verzweigungsbefehle ausgeführt. Bedingte Verzweigungsbefehle können verwendet werden, um hochstufige Programmkonstrukte aufzubauen, wie z.B. Schleifen- und Wenn-dann-sonst-Anweisungen (if-then-else-Anweisung). Da das Schleifen- und das If-then-else-Programmkonstrukt so herkömmlich sind, ist es wesentlich, daß die bedingten Verzweigungsbefehle, die diese ausführen, so effizient wie möglich ausgeführt werden.

Ein Computerbefehl wird durch Durchführen eines oder mehrerer Schritte ausgeführt. Typischerweise sind diese Schritte, erstens, den Befehl, auf den durch den Programmzähler gezeigt wird, zu holen, zweitens, die Operation, die durch den Befehl angezeigt ist, zu decodieren und durchzuführen, und abschließend, die Ergebnisse zu sichern. Ein einfacher Verzweigungsbefehl ändert den Inhalt des Programmzählers, um die Ausführung eines "Sprungs" ("jump") an einen anderen Ort innerhalb des Programms zu bewirken. Um die Ausführung von Computerbefehlen zu beschleunigen, wurde eine Technik zum

Ausführen von mehr als einem Befehl zur selben Zeit entwickelt, was als Pipeline-Verfahren bezeichnet wird. Das Pipeline-Verfahren ermöglicht es, der zentralen Verarbeitungseinheit, CPU, z.B. einen Befehl zu holen, während ein anderer Befehl ausgeführt wird, und während die Ergebnisse eines dritten Befehls zur selben Zeit gesichert werden. Bei Computerarchitekturen, die das Pipeline-Verfahren verwenden, ist das Verzweigen ein aufwendiger Betrieb, da Verzweigungsbefehle bewirken können, daß andere Befehle in der Pipeline abhängig vom Ergebnis des Verzweigungsbefehls zurückgehalten werden. Wenn ein bedingter Verzweigungsbefehl ausgeführt ist, bei dem die Bedingung wahr ist, bewirkt dies, daß die CPU mit der Ausführung bei einer neuen Adresse, die als Zieladresse bezeichnet wird, fortfährt. Nachdem in einem Computer mit dem Pipeline-Verfahren das Holen des Befehls gleichzeitig mit der Decodierung und Ausführung des Befehls erfolgt, hat der Computer bereits den Befehl geholt, der dem Verzweigungsbefehl in dem Programm folgt. Dies ist ein unterschiedlicher Befehl gegenüber dem Befehl an der Zieladresse. Deshalb muß die CPU die Befehls-Pipeline nach dem Verzweigungsbefehl anhalten, bis das Ergebnis des Verzweigungsbefehls bekannt ist, und der geeignete Befehl geholt ist. Um den Durchsatz des Computers zu maximieren, haben Computerentwickler versucht, Computer zu entwickeln, die den Durchsatz durch Minimieren der Notwendigkeit des Anhaltens der Befehls-Pipeline maximieren.

Im Stand der Technik wurden verschiedene Schemata verwendet, um das Anhalten der Befehls-Pipeline bei bedingten Verzweigungen zu vermeiden. Als erste haben einige Hochleistungsprozessoren verschiedene Verzweigungsvorhersageschemata verwendet, um abzuschätzen, ob die Verzweigung ausgeführt wird oder nicht. Dieser Ansatz erfordert eine extensive Hardware und ist aufgrund der erforderlichen, teuren Hardware für alle, außer für Computer mit höchstem Verhalten, unannehmbar. Zweitens haben andere Architekturen sowohl den Befehl in dem Programm, der der Verzweigung folgt, und den Befehl an der Verzweigungszieladresse geholt. Dieser Ansatz ist

nicht annehmbar, da er ebenfalls eine teure Hardware und zusätzliche Speicherzugriffe erfordert, um immer beide Befehle zu holen. Drittens haben einige Architekturen ein Bit in dem Befehl, um dem Computer mitzuteilen, ob es wahrscheinlicher ist, daß der Befehl, der der Verzweigung folgt, oder der Befehl an der Verzweigungszieladresse ausgeführt wird. Der Computer holt dann den wahrscheinlicheren Befehl und hält die Pipeline nur an, wenn die Abschätzung falsch ist. Dieser Ansatz erfordert eine teure Hardware, und bewirkt, wenn die Abschätzung falsch ist, daß zusätzliche Zeit aufgewendet werden muß, um die Pipeline zu sichern, und um den geeigneten Befehl zu holen. Viertens verwenden andere Architekturen zwei Bits, die die CPU auf der Grundlage, ob die Verzweigung durchgeführt wird oder nicht, anweisen, den Befehl, der dem Verzweigungsbefehl folgt, immer oder niemals auszuführen. Diese Architektur verwendet zu viele Bits des Befehls, wodurch der maximale Bereich des Verzweigungsbefehls reduziert wird. Abschließend führen wiederum andere Architekturen die Befehle in dem Programm, die dem Verzweigungsbefehl folgen, immer aus, bevor die Verzweigung durchgeführt wird oder nicht durchgeführt wird.

Die Technik der Ausführung des Befehls in dem Programm, der dem Verzweigungsbefehl folgt, ist als verzögerte Verzweigung bekannt. Eine verzögerte Verzweigung ist wünschenswert, da der Befehl in der Pipeline immer ausgeführt wird, und die Pipeline nicht angehalten wird. Dies tritt auf, da die verzögerte Verzweigung dem Computer Zeit gibt, den Verzweigungsbefehl auszuführen, und um die Adresse des nächsten Befehls zu berechnen, während der Befehl in der Pipeline ausgeführt wird. Obwohl diese Technik das Anhalten der Befehls-Pipeline vermeidet, kann sie das Anordnen eines Nicht-Operationsbefehls, der dem Verzweigungsbefehl folgt, erfordern, was das Verhalten nicht verbessern würde, nachdem der zusätzliche Speicherzugriff jede Verbesserung vermeidet.

Eine Softwaretechnik, die den Vorteil der verzögerten Verzweigung verwendet, ist das Mischen (Merger). Das Mischen

arbeitet mit Schleifenkonstrukten, bei denen der Schleifenverzweigungsbefehl am Ende der Schleife ist. Das Mischen verwendet die Vorteile der verzögerten Verzweigung durch Duplizieren des ersten Befehls der Schleife, der dem Verzweigungsbefehl der Schleife folgt, und dadurch, daß die Verzweigungszieladresse zu dem zweiten Befehl der Schleife gemacht wird. Ein mögliches Problem beim Mischen besteht darin, daß das Programm beim Verlassen der Schleife nicht notwendigerweise den verzögerten Verzweigungsbefehl wieder ausführen will. Dies ist ein Problem für Architekturen, die immer die verzögerte Verzweigung verwenden.

Wenn viele Computersysteme nach dem Stand der Technik bestimmen, daß eine Verzweigung ausgeführt werden soll, halten die Computersysteme die Befehls-Pipeline an oder verriegeln diese. Die Verriegelung der Pipeline schließt das Anhalten des Computers beim Holen des nächsten Befehls und das Verhindern, daß die Pipeline weitergeht, um irgendeinen der Befehle in der Pipeline auszuführen, ein. Die Verriegelung reduziert die Verhaltensverbesserung, die durch das Pipeline-Verfahren erreicht wird, und sollte deshalb vermieden werden.

Gemäß der Erfindung, wie sie in der Stammanmeldung EP-A-02070665 beansprucht ist, werden ein Verfahren und eine Vorrichtung zur bedingten, verzögerten Verzweigung innerhalb eines digitalen Computers geschaffen.

Die vorliegende Erfindung schafft ein Verfahren und eine Vorrichtung zur Annullierung, z.B. eines Verzögerungsschlitzbefehls, der dem Verzweigungsbefehl folgt, wenn der Verzögerungsschlitzbefehl nicht effizient verwendet werden kann.

#### Beschreibung der Zeichnungen

Fig. 1 ist ein Verzweigungsbefehl.

- Fig. 2 stellt ein Verfahren der Verzweigung dar.
- Fig. 3 ist ein Flußdiagramm des Verfahrens der Verzweigung.
- Fig. 4 ist ein Funktionsblockdiagramm einer Vorrichtung gemäß dem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung.
- Fig. 5 ist ein Zeitverlaufzustandsdiagramm der Vorrichtung in Fig. 4.

#### Beschreibung des bevorzugten Ausführungsbeispiels

Fig. 1 ist ein Verzweigungsbefehl. Der Verzweigungsbefehl 501 besteht aus 32 Informationsbits, die durch den Computer verwendet werden, um den Befehl auszuführen. Dieser Befehl verbindet die Funktion der Verzweigung mit dem Betrieb des Vergleichens von zwei Operanden. Der Befehl 501 enthält ein 6-Bit-Operationscodefeld 502, ein 5-Bit-Adreßfeld 503 eines ersten Quellenregisters, ein 5-Bit-Adreßfeld 504 eines zweiten Quellenregisters, ein 3-Bit-Bedingungscodefeld 505, ein 11-Bit-Verzweigungsverschiebungsfeld 506 und ein 1-Bit-Verschiebungsfeldvorzeichenbit 508, und ein Annullierungsbit 507. Das Betriebscodefeld 502 bezeichnet die Befehle als einen Vergleichs- und einen Verzweigungsbefehl. Die Adreßfelder 503 und 504 für das erste und das zweite Quellenregister bezeichnen die Register, deren Inhalte verglichen werden. Die Verzweigungsverschiebung, die positiv oder negativ sein kann, wird durch die Felder 508 und 506 bestimmt. Diese Verschiebung wird verwendet, um die Zieladresse für die Verzweigung zu berechnen. Der nächste Befehl in der Befehls-Pipeline kann gemäß der vorliegenden Erfindung durch Setzen des Annullierungsbits 507 annulliert werden.

Bei der vorliegenden Erfindung kann die Ausführung des derzeitigen Befehls annulliert werden. Der Zweck der Annullierung besteht darin, es so erscheinen zu lassen, als ob der

Befehl niemals in der Pipeline existierte, obwohl der Befehl geholt wurde und dessen Operation durchgeführt wurde. Die Annullierung wird durch Vermeiden, daß der Befehl irgendeinen Zustand der CPU ändert, erreicht. Um die Veränderung des Zustands des Computers zu vermeiden, muß der Annullierungsprozeß das Schreiben irgendwelcher Ergebnisse des annullierten Befehls in irgendein Register oder an irgendeinen Speicherort vermeiden, und vermeiden, daß irgendwelche Nebeneffekte auftreten, z.B. die Erzeugung von Unterbrechungen (Interrupts), die durch den annullierten Befehl bewirkt wurden. Dies wird durch Qualifizieren jedes Schreibsignals mit dem Annullierungssignal, das in dem vorhergehenden Befehl erzeugt wurde, durchgeführt, wodurch vermieden wird, daß der Befehl irgendwelche Ergebnisse irgendeiner Berechnung speichert oder auf sonstige Weise den Zustand des Computersystems ändert. Ein einfacher Weg zum Qualifizieren des Schreibsignals des derzeitigen Befehls ist die UND-Verknüpfung des Schreibsignals mit einer zurückgehaltenen Kopie des annullierten Signals, das in dem vorhergehenden Befehl erzeugt wurde. Das Annullierungssignal, das durch einen Befehl erzeugt wurde, kann z.B. in dem Prozessorstatuswort zur Verwendung in dem folgenden Befehl gespeichert sein. Die Annullierung ist eine sehr nützliche Technik, da sie es ermöglicht, einen Befehl in die Pipeline zu holen, ohne Bedenken, ob eine Entscheidung, die durch einen anderen Befehl in der Pipeline getroffen wird, bewirkt, daß dieser Befehl nicht ausgeführt werden soll. Der Befehl durchläuft einfach die Pipeline, bis die Zeit kommt, um dessen Ergebnisse zu speichern, und der Befehl kann dann in der letzten Minute mit derselben Auswirkung annulliert werden, als ob der Befehl niemals in der Pipeline vorhanden war.

Bei einem Computersystem, daß das Pipeline-Verfahren verwendet, existieren zwei unterschiedliche Konzepte bezüglich des als nächsten auszuführenden Befehls. Das erste Konzept ist ein Zeit-sequentieller Befehl, der der nächste Befehl in der Befehls-Pipeline nach dem derzeitigen Befehl ist. Dieser Befehl wird nach dem derzeitigen Befehl ausgeführt, und die

Ergebnisse der Operation werden gespeichert, wenn sie nicht annulliert werden. Das zweite Konzept ist ein Raum-sequentieller Befehl. Dies ist ein Befehl, der unmittelbar dem derzeitigen Befehl in dem Programm folgt. Im allgemeinen wird der Raum-sequentielle Befehl für den derzeitigen Befehl der Zeit-sequentielle Befehl sein. Die Ausnahme von der Regel tritt bei durchgeführten Verzweigungsbefehlen auf, bei denen der Zeit-sequentielle Befehl der Befehl an der Zieladresse ist, der im allgemeinen nicht der Raum-sequentielle Befehl des Verzweigungsbefehls ist.

Der Verzögerungsschlitzbefehl ist der Zeit-sequentielle Befehl eines Verzweigungsbefehls. Im allgemeinen wird der Verzögerungsschlitzbefehl der Raum-sequentielle Befehl des Verzweigungsbefehls sein. Die Ausnahme von dieser Regel stellt der Fall einer Verzweigung dar, die einem Verzweigungsbefehl folgt. Für diesen Fall wird der verzögerte Schlitzbefehl für den zweiten Verzweigungsbefehl die Zieladresse des ersten Verzweigungsbefehls sein, und nicht der Raum-sequentielle Befehl des zweiten Verzweigungsbefehls.

Eine nicht-bedingte Verzweigung stellt das Konzept der Annullierung und des Verzögerungsschlitzbefehls deutlich dar. Wenn das Annullierungsbit "aus" ist, wird der Verzögerungsschlitzbefehl des nicht-bedingten Verzweigungsbefehls immer ausgeführt. Dies entspricht der ständigen Verwendung der verzögerten Verzweigung. Wenn das Annullierungsbit "aus" ist, wird der Verzögerungsschlitzbefehl des nicht-bedingten Verzweigungsbefehls immer annulliert. Dies ist äquivalent zu der niemalsigen Ausführung des Verzögerungsschlitzbefehls.

Fig. 2 stellt ein Verfahren der bedingten Verzweigung dar. Ein Computer, der das Verfahren aus Fig. 2 ausführt, hat ein Programm 101, das aus Befehlen 100 besteht, die einen bedingten Verzweigungsbefehl 102 einschließen. Der Raum-sequentielle Befehl des Verzweigungsbefehls 102 ist der Befehl 103. Für einen bedingten Verzweigungsbefehl 102 mit negativer Verzweigungsverschiebung ist der Befehl 104 an der

Zieladresse. Für einen bedingten Verzweigungsbefehl 102 mit einer positiven Verzweigungsversetzung ist der Befehl 105 bei der Zieladresse. Die Ausführung des Programms ist durch die Graphen 110, 111, 112, 113 und 114 dargestellt. Während der normalen Ausführung führt das Programm den derzeitigen Befehl aus, und führt dann den Raum-sequentiellen Befehl des derzeitigen Befehls aus.

Die Graphen 110, 111 und 113 zeigen die Operation eines Verzweigungsbefehls, bei dem das Annullierungsbit "aus" ist. Dies entspricht dem "niemals eine Annullierung" oder "immer ausführen" Fall. Der Verzögerungsschlitzbefehl, der dem Verzweigungsbefehl folgt, wird immer ausgeführt, unabhängig davon, ob die Verzweigung durchgeführt wird oder nicht, und ob sie eine positive oder negative Verschiebung hat. Wenn die Verzweigungsbedingung falsch ist, fährt die Ausführung mit dem Raum-sequentiellen Befehl 103 fort, wie es im Graph 110 gezeigt ist. Wenn die Verzweigungsbedingung wahr ist, wird der Verzögerungsschlitzbefehl ausgeführt, und dann wird der Befehl an der Zieladresse ausgeführt, wie es im Graph 111 für eine negative Verzweigungsverschiebung und im Graph 113 für eine positive Verzweigungsverschiebung gezeigt ist.

Der Graph 110, 111, 112 und 114 stellt die Operation eines Verzweigungsbefehls dar, bei dem das Annullierungsbit "ein" ist. Dies entspricht dem nachfolgend beschriebenen Fall "manchmal Annullierung". Wenn das Annullierungsbit "ein" ist, kann der Verzögerungsschlitzbefehl abhängig von der Richtung der Verzweigung und abhängig davon, ob die Bedingung, die bestimmt, ob die Verzweigung durchgeführt wird oder nicht, wahr oder falsch ist, annulliert werden. Die Graphen 110 und 114 stellen die Operation des Verzweigungsbefehls dar, wenn die Bedingung, die die Verzweigung auslöst, falsch ist, was bewirkt, daß die Verzweigung nicht durchgeführt wird. Wenn die Verzweigungsverschiebung positiv ist, wird der Verzögerungsschlitzbefehl ausgeführt, wie es durch den Graph 110 gezeigt ist. Wenn die Verzweigungsverschiebung negativ ist, wird der Verzögerungsschlitzbefehl

annulliert, wie es durch den Graph 114 gezeigt ist. Die gestrichelte Linie in den Graphen 112 und 114 zeigt an, daß der Verzögerungsschlitzbefehl, obwohl er geholt wurde, annulliert werden wird, als ob er in der Befehls-Pipeline nie existierte.

Die Graphen 111 und 112 stellen die Operation des Verzweigungs-befehls dar, bei dem das Annullierungsbit "ein" ist, wenn die Bedingung, die die Verzweigung auslöst, wahr ist, was bewirkt, daß die Verzweigung durchgeführt wird. Wenn die Verzweigungsverschiebung positiv ist, wird der Verzögerungsschlitzbefehl annulliert, wie es im Graph 112 gezeigt ist, und die Ausführung fährt bei der Zieladresse fort. Wenn die Verzweigungsverschiebung negativ ist, wird der Verzögerungsschlitzbefehl ausgeführt, wie es im Graph 111 gezeigt ist, bevor an der Zieladresse fortgefahren wird.

Fig. 3 ist ein Flußdiagramm des Verfahrens der Verzweigung. Die Graphen 111 bis 114 werden unter Bezugnahme auf das Flußdiagramm besser verstanden. Der erste Schritt besteht darin, zu bestimmen, ob das Annullierungsbit "ein" ist. Wenn das Annullierungsbit "aus" ist, dann wird der Verzögerungsschlitzbefehl für den Verzweigungsbefehl immer ausgeführt. Dies tritt auf, abhängig davon, ob die Verzweigung durchgeführt wird oder nicht. Wenn das Annullierungsbit "ein" ist, dann wird der Verzögerungsschlitzbefehl, der der Verzweigung folgt, nicht ausgeführt, außer die Verzweigung wird durchgeführt und die Verzweigungsverschiebung ist negativ, oder die Verzweigung wird nicht durchgeführt, und die Verzweigungsverschiebung ist positiv.

Die Operation führt ein sehr einfaches, aber effektives Verfahren der statischen Verzweigungsvorhersage aus, daß vorhersagt, ob die Verzweigung durchgeführt wird oder nicht, und daher, welcher Befehl zu holen ist, basierend darauf, wie eine positive und eine negative Verschiebungsverzweigung durchgeführt wird. Dessen Effektivität hängt von der Computersoftware ab, die einem Satz von Softwarekonventionen beim

Ausführen bestimmter höherstufiger Programmsteuerungskonstrukte durch einen bedingten Verzweigungsbefehl folgt. Ein Schleifenkonstrukt wird z.B. durch eine rückwärts gerichtete bedingte Verzweigung ausgeführt, so daß ein Verzweigungsbefehl mit einer negativen Verschiebung häufig durchgeführt wird. Tatsächlich wird er von N-malen für eine Schleife, die N-male ausgeführt wird, N-1 mal durchgeführt. Ein weiteres Beispiel der angenommenen Softwarekonventionen ist, daß ein if-then-else-Konstrukt durch eine nach vorne gerichtete Verzweigung zu einem selten durchgeführten Teil ausgeführt ist, was es ermöglicht, daß der häufiger ausgeführte Teil unmittelbar nach dem Verzweigungsbefehl in dem Nicht-durchgeführte-Verzweigung-Weg liegt. Die nach vorne gerichtete Verzweigung kann z.B. eine Fehlerhandhabungsroutine sein, die in einem normalen Programm selten ausgeführt wird. Das Ausführungsbeispiel der vorliegenden Erfindung mit einem Annullierungsbit verallgemeinert und optimiert die Verwendung des Verzögerungsschlitzbefehls in Verbindung mit der oben beschriebenen statischen Verzweigungsvorhersagetechnik. Wenn das Annullierungsbit "ein" ist, bewirkt eine nach hinten gerichtete bedingte Verzweigung, die durchgeführt wird, oder eine nach vorne gerichtete bedingte Verzweigung, die nicht durchgeführt wird, was die Ziele sind, von denen durch die statische Verzweigungsvorhersagetechnik vorhergesagt wird, häufig zu sein, daß der Verzögerungsschlitzbefehl ausgeführt wird. Daher können einige nützliche Befehle in dem häufigen Weg ausgeführt werden, als der Verzögerungsschlitzbefehl, der z.B. oben anhand der Mischertechnik beschrieben wurde. Wenn das Annullierungsbit "ein" ist, bewirkt eine nach hinten gerichtete bedingte Verzweigung, die nicht durchgeführt wird, oder eine nach vorne gerichtete bedingte Verzweigung, die durchgeführt wird, die die Ziele sind, von denen vorhergesagt wird, selten zu sein, daß der Verzögerungsschlitzbefehl annulliert wird. Daher tritt die Annullierung, die das Verhalten reduziert, nur in den seltenen Fällen auf.

Wenn das Annullierungsbit "aus" ist, wird der Verzögerungsschlitzbefehl immer ausgeführt. Dies entspricht dem Fall, in

dem ein Befehl, der sowohl für den durchgeführten Verzweigungs- als auch für den nicht-durchgeführten Verzweigungsweg gemeinsam ist, als der Verzögerungsschlitzbefehl bezeichnet werden kann.

Fig. 4 ist ein Funktionsblockdiagramm einer Vorrichtung gemäß dem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung. Die Vorrichtung enthält sechs Funktionselemente: einen Befehlsspeicher 301, eine optionale virtuelle Adreß-Übersetzungseinheit 302, eine Befehlseinheit 303, eine Ausführungseinheit 304, eine optionale Gleitkomma-Einheit 305 und eine optionale Registerdatei 306. Diese Funktionselemente sind miteinander durch fünf Busse verbunden: einen Ergebnisbus 310, einen Erster-Operand-Bus 311, einen Nächster-Befehl-Bus 312, einen Zweiter-Operand-Bus 313 und einen Adreß-Bus 314. Lediglich die Ausführungseinheit 304 und die Befehlseinheit 303 sind bei der Durchführung des Betriebs des bevorzugten Ausführungsbeispiels der vorliegenden Erfindung betroffen. Die Ausführungseinheit erzeugt und/oder speichert die Bedingungen, aufgrund derer die Entscheidung zu verzweigen oder nicht zu verzweigen, getroffen wird. Die Befehlseinheit führt die Verzweigung durch Erzeugen der Adresse des nächsten Befehls, der aus dem Speicher zu holen ist, durch und schafft eine Einrichtung zum Speichern der Adresse in dem Programmzähler. Bei dem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung ist die Speichereinheit ein Hochgeschwindigkeits-Cache-Speicher, mit einer Geschwindigkeit in der Größenordnung der in der Ausführungseinheit verwendeten Logik-Schaltung.

Fig. 5 ist ein Zeitverlaufzustandsdiagramm der Vorrichtung aus Fig. 4. Das Zeitverlaufdiagramm stellt vier Stufen dar, die bei der Ausführung der Befehle 401, 402, 403 und 404 betroffen sind. Die Zeitlinie 460 ist in Stufen unterteilt, wobei die Zeit nach rechts aufgetragen ist. Die vier Zeitverlaufsstufen für jeden Befehl sind: eine Befehlsadreßerzeugungsstufe 410, eine Befehlsholsstufe 411, eine Ausführungsstufe 412 und eine Schreibstufe 413. Die Ausführung der

Befehle kann auf irgendeiner erwünschten Tiefe im Pipeline-Verfahren ausgeführt werden. Das bevorzugte Ausführungsbeispiel der vorliegenden Erfindung enthält eine vierstufige Pipeline. Wie es in Fig. 5 gezeigt ist, werden die vier Befehle gleichzeitig ausgeführt. Zum Zeitpunkt 450 ist die Schreibstufe des Befehls 401 mit der Ausführungsstufe des Befehls 402, die Befehlsholstufe des Befehls 403 und die Befehlsadreßerzeugungsstufe des Befehls 404 überlappt. Dies bedeutet für einen Verzweigungsbefehl, daß der nächste Befehl geholt werden wird, während der Verzweigungsbefehl in der Ausführungsstufe ist. Während der Befehlsadreßerzeugungsstufe wird die Adresse des nächsten Befehls aus dem Programmzähler, der die Adresse des nächsten auszuführenden Befehls enthält, berechnet und in der Befehlseinheit 303 angeordnet. Während der Befehlsholstufe wird der nächste Befehl aus dem Befehlsspeicher 301 geholt. Dies erfolgt durch Anlegen des Inhalts der Adresse, die in der Befehlsadreßerzeugungsstufe berechnet wurde, an den Adreßbus 314 und durch Übertragen des Inhalts der Adresse an den nächsten Befehlsbus 312 durchgeführt, wo er durch die Befehlseinheit decodiert wird. Der Verzweigungsbefehl kann mit anderen Operationen, z.B. einer Vergleichsoperation, kombiniert sein, die in der Ausführungseinheit 304 zu diesem Zeitpunkt ebenfalls decodiert und durchgeführt würde.

In der Ausführungsstufe 412 wird der Verzweigungsbefehl ausgeführt. Während der Ausführungsphase 412 werden sowohl die Zieladresse des Verzweigungsbefehls als auch die Adresse des Raum-sequentiellen Befehls des Verzweigungsbefehls erzeugt. Wenn der Befehl zu diesem Zeitpunkt mit einer weiteren Operation kombiniert ist, wird diese Operation durchgeführt. Am Ende der Ausführungsphase wird eine der zwei Adressen in den Programmzähler übertragen. Welche Adresse an den Programmzähler übertragen wird, wird durch die Bedingung bestimmt, die in der Ausführungseinheit 304 gespeichert ist. Während der Schreibphase 413 tritt keine Operation auf, außer wenn ein Ergebnis eines kombinierten Befehls gespeichert werden muß. Durch Durchführen des Schreibens der Ergebnisse in

einen Speicher oder in die Register und irgendwelcher Seiteneffekte, wie z.B. einer Unterbrechungsbestätigung, die durch einen Befehl bewirkt wurde, nicht vor der Stufe 412 und 413, ermöglicht dieser Ansatz eine einfachere Ausführung des Konzepts der Annullierung eines Befehls, der immer in der Pipeline ist.

Europäisches Aktenzeichen: 90203019.6-2201

Patentansprüche

1. Ein Verfahren zum Annullieren eines Befehls, der eine Operation durchführt und fähig ist, Ergebnisse, Fehler, nicht-programmierte Programmsprünge und Unterbrechungen in einem Computersystem mit Speicher, daß das Pipeline-Verfahren verwendet, zu erzeugen, wobei das Verfahren folgende Schritte aufweist:

Holen eines ersten und eines zweiten Befehls aus dem Speicher in die Befehls-Pipeline;

Durchführen der Befehle, die durch den ersten und den zweiten Befehl angezeigt sind, und

Vermeiden, daß irgendwelche Ergebnisse, Fehler, nicht-programmierte Programmsprünge oder Unterbrechungen, die durch das Holen oder Durchführen der Operation, die durch den zweiten Befehl angezeigt ist, erzeugt werden, in dem Computersystem gespeichert werden oder den Betrieb des Computersystems beeinflussen,

dadurch gekennzeichnet,

daß der erste Befehl ein Annullierungsfeld hat, das mit dem Ergebnis der Operation, die durch den ersten Befehl angezeigt ist, gespeichert ist, und

daß vermieden wird, daß die Ergebnisse, Fehler, nicht-programmierten Programmsprünge oder Unterbrechungen als eine Bedingung des Zustands des Annullierungsfeldes des ersten Befehls gespeichert werden.

2. Eine Vorrichtung, um es einem ersten Befehl in einem Computersystem mit einem Annullierungssignal, das ent-

weder einen wahren oder einen falschen Zustand hat, zu ermöglichen, einen zweiten Befehl abhängig vom Zustand des Annullierungssignals zu annullieren, mit dem ein Schreibsignal die Ergebnisse des zweiten Befehls in dem Computer speichert oder Fehler, nicht-programmierte Programmsprünge und Unterbrechungen in dem Computersystem erzeugt, wobei die Vorrichtung folgende Merkmale umfaßt:

eine Einrichtung zum Zurückhalten des Zustands des Annullierungssignals nach der Ausführung des ersten Befehls; und

eine Einrichtung zum Qualifizieren des Schreibsignals des zweiten Befehls mit dem zurückgehaltenen Zustand des Annullierungssignals, um zu vermeiden, daß Ergebnisse der Ausführung des zweiten Befehls in dem Computersystem gespeichert werden, oder daß Fehler, nicht-programmierte Programmsprünge oder Unterbrechungen den Betrieb des Computersystems beeinflussen.

Europäisches Aktenzeichen: 90203019.6-2201

FIG 1

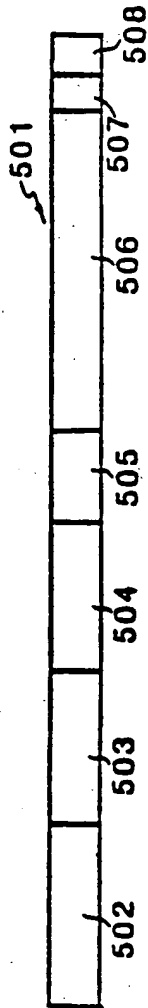


FIG 2

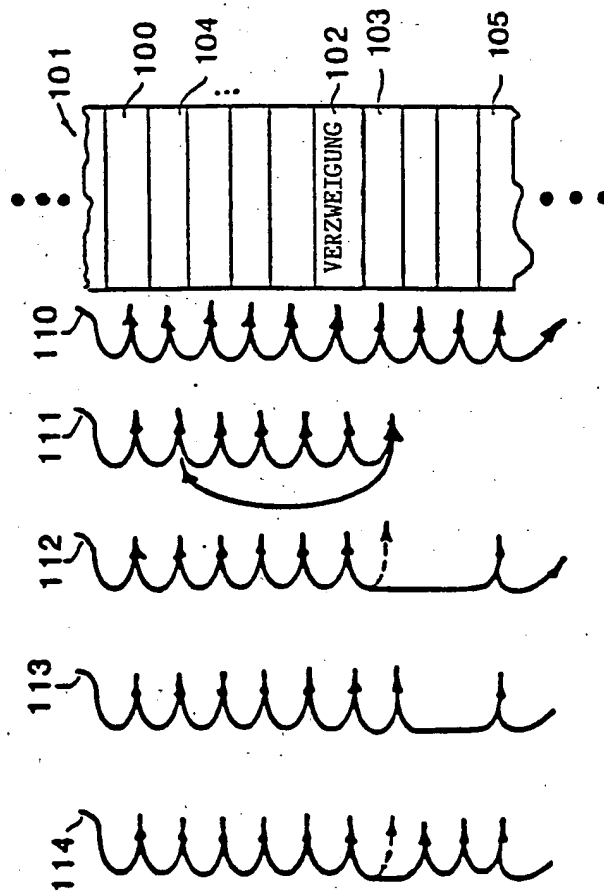
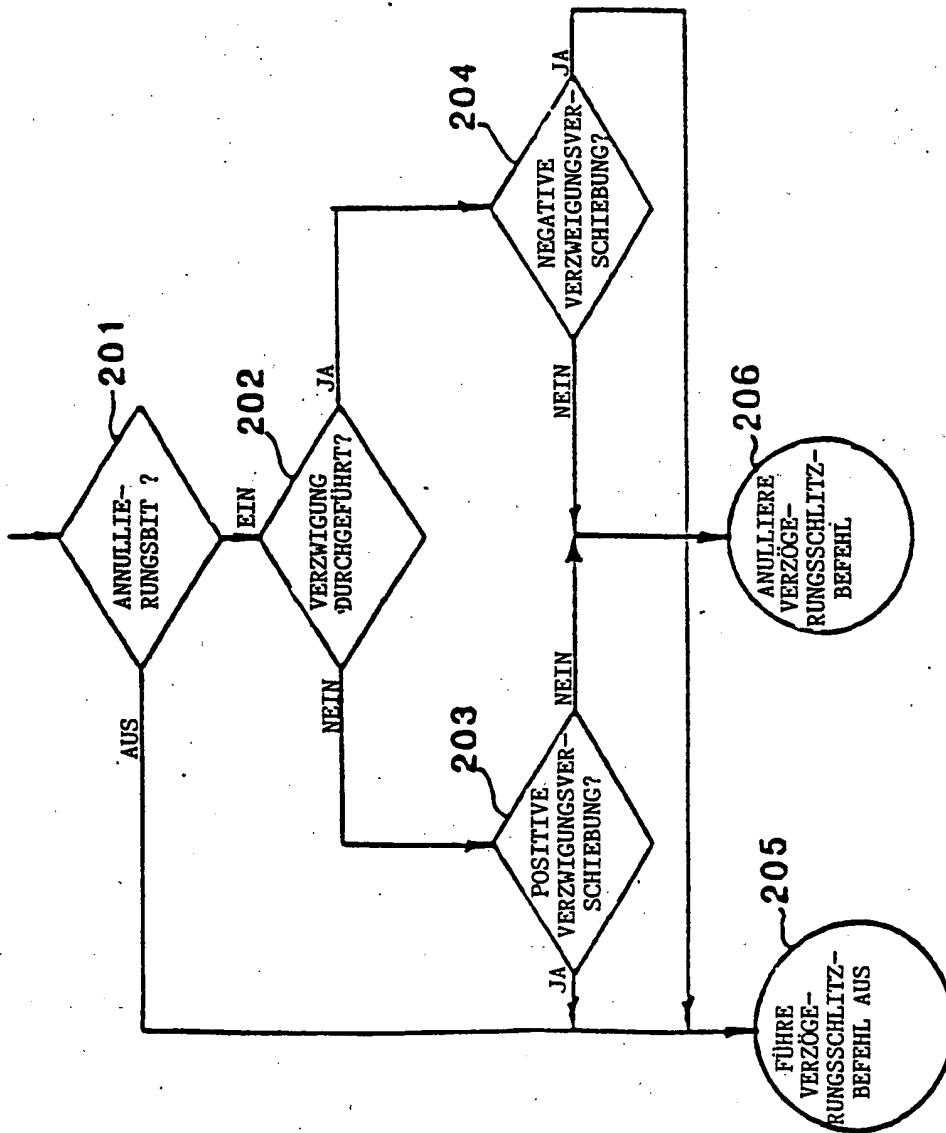


FIG 3



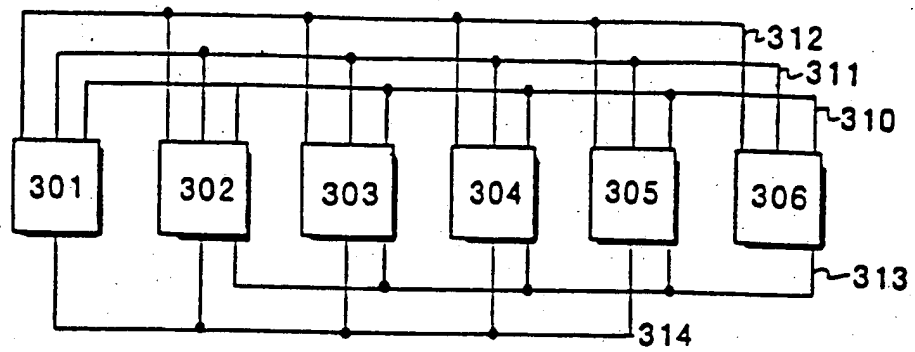


FIG 4

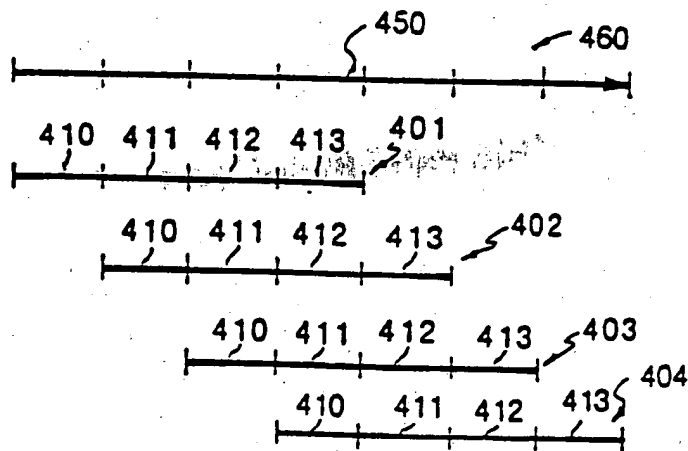


FIG 5

**THIS PAGE BLANK (USPTO)**